**Informe mongoDB**

**Repositori amb el codi:**

<https://www.github.com/nilde/GABD>

**1. Informació important:**

Per a poder executar l’script s’hauria d’inserir els fitxers proporcionats per els professors a la carpeta IMAGES, i ja es podria executar tot correctament. Abans d’executar el contingut dels script en python, s’ha de realitzar l’execució de l’script en bash tal com s’explica a l’apartat 5 d’aquesta memòria.

**2.Informació general sobre el script:**

L’script està estructurat en 3 parts molt diferenciades:

**2.1. Nexe**

És el punt central del nostre script es fa la cirda a aquest fitxer i ja s’encarregarà de fer totes les gestions necessaries.

El fitxer es el main.py.

Exemple de trucada al script:

python main.py —drop —db VECT

—drop assegura que la base de dades estigui neta abans d’inserir les noves dades.

—db VECT crea una nova base de dades amb el nom VECT

**2.2. Fitxers d’extracció:**

Son un conjunt de mòduls diferents, on cadascun s’utilitza per a poder extreure les dades automàticament de tots els fitxers que contenen les dades que s’han d’inserir. Cadascun dels fitxers al tenir formats diferents, necessiten diverses tècniques per a poder realitzar les insercions en el format correcte

Els fitxers són:

ExtractDataData

ExtractDataImages

ExtractDataNames

ExtractDataResults

**2.3. Fitxers d’inserció:**

Son un altre conjunt de petits mòduls, que s’encarreguen de configurar les dades tal que es puguin insertar correctament en els documents de la base de dades. És aquí on hi ha la configuració implícita del format dels documents.

Els fitxers són:

MakeInsertionsToDatabaseData.py

MakeInsertionsToDatabaseImages.py

MakeInsertionsToDatabaseMethods.py

MakeInsertionsToDatabaseNames.py

MakeInsertionsToDatabaseResults.py

**2.4. Fitxer de creació dels usuaris:**

Fitxer que conté una assignació de diferents rols.

El fitxer és el MakeInsertionsUsers.py

**2.5. Fitxer de configuració:**

També hi ha un fitxer de configuració que permet modificar certes parts de la generació de la base de dades.

El fitxer és el configFile.py

Estructura de la base de dades:

La base de dades té un nom variable, que contindrà tots els documents generats.

Els documents generats són:

1. DATABASE\_INFO

Conté tota la informació sobre les bases de dades,

B. RESULTS

Contindrà tota l’informació relacionada amb la generació dels resultats.

C. SET\_UP\_EXPERIMENTS

Contrindrà tota l’informació associada a les configuracions dels diferents experiments.

D.METHODS

Contindrà tota l’informació associada als diferements mètodes d’evaluació que s’utilitzen en la detecció d’outliers

E. EXPERIMENTS

Contindrà tota l’informació relacionada amb la generació dels experiments.

F. VECTORS

Contindrà tota l’informació associada amb els vectors de les diferents bases de dades importades de l’UCI.

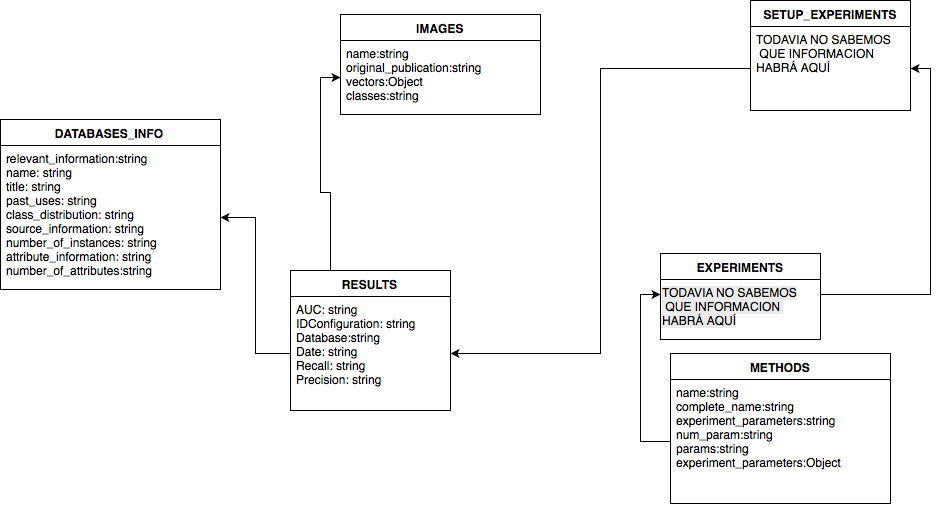
G. IMAGES

Contindrà tota l’informació de les dades associades a les diferents imatges que s’han de classificar, tant les seves imatges com els vectors de caraterístiques que estàn associats.

**2.6 Observacions en la generació:**

En la creació de la base de dades, ens hem asonat de que l’estructura que generem té un impacte molt significatiu en el rendiment que obtenim després. Si obtem per des

**3. Diagrama de classes de la base de dades:**



**4. Tutorial execució dels scripts de bash:**

1. Obrir el terminal

2. Situarse al directori (directori actual + /GABD/mongoDBScripts/consoleScripts)

3. Executar la comanda chmod 700 scriptMongo.js

4. Executar la comanda mongo < scriptMongo.js

El petit script aquest, permet utilitzar tamanys del chunk personalitzats. Està posat a 1024MB per a poder permetre que càpiguen totes les dades necessaries.

**5. Exercicis de Seguretat de BD: creació d’usuaris i drets d’accés (associats a mongo)**

***1. Feu el disseny dels perfils, rols i privilegis necessaris per a implementar els usuaris descrits en la secció anterior. Per a cada usuari definiu un primer conjunt de privilegis segons les necessitats que preveieu que pugueu tenir. Com a mínim el conjunt de privilegis que haurà de tenir cada usuari són:***

***Detector d’outliers: consultes a les taules o col·leccions.***

***Gestor de Dades: operacions de lectura, actualització i esborrat de dades.***

***Desenvolupador: creació, modificació i esborrat de taules o col·leccions.***

***Auditors: veure els objectes associats al projecte i connexions actives.***

***Més endavant haureu d’ampliar els privilegis per algun dels usuaris que heu definit. Ho haureu de fer i justificar arribat el moment.***

Aquesta creació dels usuaris s’ha realitzat al fitxer **makeInsertionUsers**, on hem utilitzat simplement rols per defecte que té mongoDB.L’implementació d’aquestes creacions es troben de la línia 9 a la 13 i necessitem utilitzar el client que s’utilitza en la creació i inserció de les dades.

***2. Feu un script que mostri, per a cada usuari i rol que hageu creat, els privilegis assignats a cadascun d’ells. En cas que creeu algun perfil, el script que creeu també ho haurà de mostrar així com els paràmetres que hagueu configurat. Consulteu la documentació associada a aquesta sessió per obtenir un llistat d’instruccions, taules i vistes on buscar la informació que es demana. Aquest script s’ha d’executar des de l’usuari que faci les funcions de DBA en cada SGBD.***

Per a comprovar quins son els usuaris associats a la BD de dades, ho hem realitzat de dues maneres diferents, la primera es mitjançant l’execució del script que inserta les dades i un cop surt la consola interactiva seleccionar l’opció usersinfo, i retornarà una llista completa dels usuaris que hi han a la bd. La segona es mitjançant un microscript que es diu showUsers i on l’execució es fa mitjançant la comanda mongo < showUsers.js,(pot ser necessari donar permisos d’execució chmod +x, o bé chmod 700).

***3. Feu un script de validació que creï un joc de proves que posi en evidència els rols/privilegis assignats a cada usuari. ). Aquest script haurà de crear una taula Test amb dos camps: camp1 i camp2 amb tipus varchar2 i number, respectivament. A continuació haurà d’inserir almenys 2 registres, modificar-ne un i esborrar un altre. També haurà de fer una consulta. Aquest script s’ha d’executar des de l’usuari que faci les funcions de DBA. Dins el script, s’haurà de canviar d’usuari per comprovar que pot fer, i que no pot fer, cadascun un d’ells.***

*Per a poder permetre la connexió amb autenticació, el que s’ha de fer primer es generar tots els rols que es vulguin i després engegar el dimoni de mongo amb els següents paràmetres mongod --auth --port 27017 --dbpath /data/db. Això permetrà la connexió mitjançant autenticació.*

***TODO***

***6.Exercici Nivell intern dels SGBD***

***4. [mongoDB] Indiqueu a quin directori es troben els fitxers de col·leccions que conté mongoDB. Indiqueu també el nom dels fitxers que conté i comproveu si hi ha alguna modificació si intentem crear una nova col·lecció. Investigueu si podem trobar la mida de les col·leccions des de la base de dades. Si és així genereu el script que ho evidenciï.***

*Per saber quin path estem utilitzant, quan correm el dimoni de mongo, el que hem de fer es utilitzar una opció concreta que permet especificar un path concret on es trobaran allotjats tots els fitxers relacionats amb la base de dades que estarà en execució.*

*El nom dels fitxers que contenen las col·leccions són tots aquells que contenen el prefix collections dintre de la carpeta. També hi ha un fitxer \_mdb\_catalog.wt que conté informació genèrica sobre el contingut total de la base de dades.*

*A més a més, conté una sèrie de fitxers d’index que indiquen tots els índex associats a les col·leccions. També hi han més fitxers tots relacionatts amb la supervisió de la base de dades.*

*La mida de les col·leccions de la base de dades, la podrem trobar mitjançant la connexió amb un client gràfic directament o mitjançant la comanda db.stats.*

*L’script que conté aquesta prova es el showDatabaseInfo.*

*Per a poder comprobar el correcte funcionament millor primer executar el script de test amb la comanda,* ***mongo < testUsers.js*** *i després* ***mongo < showDatabaseInfo.js***

***5. [Oracle i mongoDB] Feu la inserció de les dades utilitzant els scripts en python que us proporcionem2. Dels scripts només heu de modificar la part on es fan els inserts (ja que els heu d’adaptar a l’estructura de taules que heu dissenyat) així com els paràmetres de connexió. Feu la inserció de les dades amb el usuari de tipus Desenvolupador. Les funcions, o procediments, que haureu d’editar són: insertVectorDataset, insertImageDataset i insertDescriptors i que trobareu als fitxers: oracleConnection i mongoConnection. Important!! Aspectes a tenir en compte:***

***ii) [Oracle i mongoDB] No confirmeu cap transacció fins que no estiguin fetes totes les insercions de cada una de les funcions python que heu modificat.***

*Això, es pot fer mitjançant el script main.py contingut en la carpeta GABD/mongoDBScripts amb la comanda* ***python main.py —drop —db VECT***

***6. [Oracle i mongoDB] Feu les consultes que calguin per demostrar que heu inserit correctament les dades del exercici 5.***

*Això es fa mitjançant l’script checkInsertions amb la comanda* ***mongo < checkInsertions.js,*** *s’ha de tenir en compte que primer s’ha de realitzar la inserció de les dades anteriors.*

*De moment únicament es retornan tots els elements ja que ja serveix per a mirar si les dades s’han inserit correctament.*